

Technická univerzita v Liberci

Fakulta přírodovědně-humanitní a pedagogická

Katedra: Informatiky

Studijní program: 2. stupeň

Kombinace: matematika-informatika

TVORBA APLIKACE NA GENEROVÁNÍ
ROZVRHŮ HODIN PRO ZŠ

CREATION OF AN APPLICATION FOR
MAKING ELEMENTARY SCHOOL TIME-
TABLES

Autor:

Jan Cehák

Podpis:

.....

Adresa:

Mírová 541

289 22, Lysá nad Labem

Vedoucí práce: Ing. Petr Kretschmer

Konzultant: Ing. Igor Kopetschke

Počet

stran	slov	obrázků	Tabulek	pramenů	příloh
52	7474	25	3	1	3

V Liberci dne: 10.4.2010

Katedra:

ZADÁNÍ DIPLOMOVÉ PRÁCE

(pro magisterský studijní program)

pro (diplomant)

adresa:

obor (kombinace):

Název DP:

Název DP v angličtině:

Vedoucí práce:

Konzultant:

Termín odevzdání:

Pozn. Podmínky pro zadání práce jsou k nahlédnutí na katedrách. Katedry rovněž formulují podrobnosti zadání. Zásady pro zpracování DP jsou k dispozici ve dvou verzích (stručné.resp. metodické pokyny) na katedrách a na Děkanátě Fakulty pedagogické TU v Liberci.

V Liberci dne

.....

děkan

vedoucí katedry

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci dne: 10.04.2010

Jan Cehák

Poděkování

Děkuji své manželce Kateřině, která vydržela v posledních několika měsících moji „duševní nepřítomnost“, kdy jsem o volných chvílích před a po práci zasedl k počítači a dlouhé hodiny programoval. Děkuji svým rodičům Markétě a Jiřímu Cehákovým, kteří mě podporovali celá studia jak finančně, tak morálně a bez nichž bych nyní doslova nebyl tam, kde jsem. Dále bych poděkoval mému příteli Janu Vojtěchovi, který jako odborník na programovací jazyk C++ snášel řadu mých, někdy i vyloženě hloupých, otázek a vždy se mi snažil co možná nejsrozumitelněji odpovědět. V neposlední řadě bych rád poděkoval vedoucímu mé diplomové práce Ing. Petru Kretschmerovi, který mě při tvorbě programu a diplomové práce směřoval tím správným směrem.

Anotace

Diplomová práce Tvorba aplikace na generování rozvrhů hodin pro ZŠ, pracovní název „Deroz“ – rozvrh hodin, je rozsáhlým projektem, na kterém by se dalo soustavně pracovat dalších mnoho měsíců. Program je vytvářen v C++ Builder. Program v současné verzi zahrnuje tři fáze tvorby rozvrhu: zadávání dat, potřebných nejen k tvorbě rozvrhu, generování rozvrhu, kdy se do rozvrhů zadá až 90 % všech předmětů a následné ruční dotvoření rozvrhů, pod čímž se skrývá mazání a zadávání předmětů. Hlavního cíle, vytvoření algoritmu na tvorbu rozvrhu, se podařilo dosáhnout s využitím tří základních nutných a dvou přidavných volitelných podmínek. Pro program existuje mnoho dalších možných rozšíření, které lze aplikovat v budoucích verzích.

Annotation

The diploma paper “Creation of an Application for Making Elementary School Time-Tables“, in another way called “Deroz - the time-table”, is a large project which could be worked on for many more months. The programme was created in C++ Builder software. The programme contains three stages of time-table-making at the moment. These are: data entry, which are not only necessary for creation of a time-tables, generating of time-tables, when as many as 90% of school subjects are placed and finally manually operated completion of the time-table, which consists of deleting and replacing subject. The main goal was fulfilled by usage of three basic compulsory and two added optional conditions. There are many more possible upgrades existing for this programme that could be applied in its future versions.

Klíčová slova

Rozvrh hodin, aplikace, generování, algoritmus

Keywords

Time-table, application, generating, algorithm

Obsah

Prohlášení.....	3
Poděkování.....	4
Anotace.....	5
Annotation.....	5
Klíčová slova.....	5
Keywords.....	5
Obsah.....	6
1. Úvod.....	8
2. Zadávání dat.....	10
2.1 Učebny.....	10
2.2 Předměty.....	11
2.3 Učitelé.....	11
2.4 Žáci.....	13
2.5 Třídy.....	14
2.6 Učební plán.....	15
2.7 Skupiny.....	15
2.8 Nutná data pro algoritmus na tvorbu rozvrhu.....	16
2.9 Nepovinná data, nutná pro jednoduché rozšíření programu.....	16
3. Testovací data.....	17
4. Optimalizace.....	18
5. Generování rozvrhu.....	19
5.1 Základní úkoly algoritmu.....	19
5.2 Postup generování rozvrhu.....	20
5.2.1 První fáze.....	20
5.2.2 Druhá fáze.....	20
5.2.3 Třetí fáze.....	21
5.2.4 Čtvrtá fáze.....	21
5.2.5 Pátá fáze.....	21
5.2.6 Šestá fáze.....	21
5.3 Další možné podmínky při generování rozvrhu.....	23
6. Manuální úpravy rozvrhu.....	24
6.1 Odebírání předmětů z rozvrhu.....	24
6.2 Přidávání předmětů do rozvrhu.....	24
7. Možná vylepšení programu.....	25
7.1 Zadávání informací.....	25
7.2 Zadávání informací o učebnách.....	25
7.3 Zadávání informací o předmětech.....	25
7.4 Zadávání informací o učitelích.....	25
7.5 Zadávání informací o třídách.....	25
7.6 Konečné návrhy na vylepšení.....	26
8. Uživatelská příručka.....	27
8.1 Instalace programu.....	27
8.2 Úvodní stránka.....	28
8.3 Soubor.....	29
8.4 Zadávání dat.....	30
8.4.1 Tlačítka a funkce při zadávání dat.....	30
8.4.2 Učebny.....	33
8.4.3 Předměty.....	34

8.4.4 Učitelé	35
8.4.5 Žáci.....	37
8.4.6 Třídy	38
8.4.7 Skupiny.....	39
8.4.8 Učební plán.....	40
8.5 Generování rozvrhu	41
8.6 Konečné úpravy v rozvrhu	43
8.6.1 Třídy	43
8.6.2 Učitelé	47
8.6.3 Učebny.....	48
Závěr	49
Literatura.....	50
Příloha 1: Seznam obrázků	51
Příloha 2: Seznam tabulek	51
Příloha 3: Obsah CD.....	51

1. Úvod

Již od dětství jsem se s tvorbou rozvrhu setkával doma, kde moje matka každé letní prázdniny rozvrhy vytvářela. Setkal jsem se tedy s několika metodami, jak rozvrh vytvářet. První, nejnáročnější metodou, kterou jsem mohl pozorovat, bylo zasouvání barevných, popsaných čtverečků o rozměrech 1 x 1 cm vyrobených ze čtvrtky do dvou nástěnek o rozměrech 1,5 x 1,5 m. Zadávání každého předmětu tehdy provázelo dlouhé a důkladné zkoumání, zda se předmět nekryje s jiným, zda vyučující v danou dobu neučí i v jiné třídě, zda pro předmět existuje volná místnost a neustále se musel kontrolovat počet již zadaných předmětů u každé třídy, aby nebylo překročeno hodinové dotace jednotlivých předmětů. Po umístění všech čtverečků symbolizujících předměty nastala dlouhá fáze přepisování do třech typů rozvrhů, které jsou potřebné pro úspěšný start školního roku - rozvrh hodin třídy, učitelů a učeben. Při této zdlouhavé a fyzicky náročné práci docházelo často k omylům a chybám.

Po několika letech, když se začínaly ve větší míře používat počítače, se objevily i první programy na tvorbu rozvrhů. První, se kterým jsem se setkal, byl psán v jazyce Pascal a na jeho vývoji pracoval souběžně s jeho používáním v praxi jeden programátor několik let. Neustálé testování a ladění chodu programu bylo na denním pořádku. Další program, který moje matka používá dodnes je již mnohem více propracovaný a jeho součástí je již i suplování, tisk rozvrhů, databáze známek, databáze žáků, docházka a podobné funkce, které pomáhají při správě chodu školy. Na tomto programu již pracoval tým programátorů a nedovolím si odhadnout, jak dlouho trval jeho vývoj. Díky tomu, že i tento program není ideální a algoritmus, který vytváří rozvrh, nedokáže vytvořit bezchybný rozvrh hodin pro celou školu, rozhodl jsem se naprogramovat „Deroz – rozvrh hodin“ a vyzkoušet si obtížnost samotného algoritmu na vlastní kůži. Tento robustní projekt jsem začínal s hlavou plnou nápadů na funkce, které by program měl mít. Prvotní databázi informací, které se v dalších fázích tvorby programu používají, jsem tedy futuristicky zvětšil tak, abych mohl následně všechny funkce, které jsem si vymyslel dotáhnout do zdárného konce. Během následujících týdnů, kdy jsem vytvářel samotný algoritmus, jsem ale pochopil, jak moc robustní program ve skutečnosti je. S každou další funkcí, kterou jsem v programu použil, vyvstalo vždy hned několik nových situací, které bylo potřeba ošetřit a přepracovat celé pasáže v programu, o kterých jsem si již dávno myslel, že jsou bezchybné a dokonalé. Program má tři základní fáze, které jsem musel postupně splnit. První fází je zadávání dat pro tvorbu rozvrhu a jejich ukládání do databáze. Druhou fází je generování rozvrhu, kde se má automaticky vytvořit

rozvrh pro co možná nejvíce předmětů a poslední fází je manuální úprava rozvrhu, kde se rozvrh upraví podle dalších kritérií, které nebyly použity při generování. Následují další „skryté“ fáze tvorby jakými jsou testování, grafická úprava, tvorba zkušebního rozvrhu.

2. Zadávání dat

Veškerá data, která jsou potřebná pro správné vytváření rozvrhu, jsou zadávána a ukládána pomocí dynamických jednosměrných seznamů. Jednotlivé prvky seznamu se tedy vytvářejí za chodu programu a nezabíráme tím zbytečně paměť počítače. Potřebuje-li program umístit další prvek například do seznamu tříd, alokuje se nová paměť pro nové proměnné a dosadí se do ní.

První fází tvorby rozvrhu je zadávání dat, pomocí kterých se poté může rozvrh vytvářet. Jde zde jak o data povinná, která jsou nutná pro samotnou tvorbu rozvrhu, tak i o nepovinná, která slouží k dalším vylepšením programu a k doplňujícím informacím o jednotlivých prvcích, které jsou při tvorbě potřebné. V programu Deroz – Rozvrh hodin vytvářím několik struktur pro vstupní data.

2.1 Učebny

Jde o označení, název, kapacitu a specifikaci předmětů. Pomocí těchto čtyř proměnných lze jasně definovat učebnu pro další použití v programu.

Kód:

```
struct Ucebna {  
char *a1; //zkratka  
char *a2; //nazev ucebny  
char *a3; //kapacita  
char *a4; //specializace  
Ucebna *next;  
}
```

Proměnná a1 značí zkratku učebny. V programu používám označení složená ze tří znaků. Např. A01, A02, B01. Tato proměnná je povinná u každého záznamu učebny. Proměnná a2 značí název učebny. Může to být libovolný text o maximální délce 200 znaků a slouží pro upřesnění typu učebny (např. laboratoř chemie). Tato proměnná je nepovinná. Proměnná a3 značí kapacitu učebny. Jde o maximální počet studentů, kteří mohou ve stejnou chvíli v učebně být. Proměnná není povinná, ale je důležitá pro případ, že bych při tvorbě rozvrhu tuto limitaci využíval. Více informací v kapitole „Možná vylepšení programu“. Proměnná a4 značí specializaci jednotlivých učeben. Zadává se zde výpis předmětů, které se

v dané učebně dají vyučovat. Př. tělocvična bude v tomto poli mít pouze TV, ale učebna, ve které lze vyučovat libovolný nespecializovaný předmět, zde bude mít výčet předmětů oddělený čárkou (např. MA, CJ, ZE, BI, DE). Tato proměnná je u každé položky povinná.

2.2 Předměty

Jde o zkratku a celý název předmětu, typ povinný či nepovinný a výčet vhodných učeben, kde lze předmět vyučovat.

Kód:

```
struct Predmet{  
char *a1; //zkratka  
char *a2; //nazev predmetu  
char *a3; //typ  
char *a4; //vhodne ucebny  
Predmet *next;  
};
```

Proměnná a1 zde označuje zkratku předmětu. Používám zde označení o délce 2-3 znaky (např. MA, CJ, SPE). Tento prvek je povinný. Proměnná a2 označuje celý název předmětu (např. Matematika). Tento prvek je nepovinný a slouží pouze pro upřesnění názvu. Zapotřebí je například u půlených hodin, kdy se třídy rozdělují na začátečníky a pokročilé. Pomocí označení AJ1 a AJ2 nelze zcela intuitivně zjistit, která z těchto zkratk označuje začátečníky a která pokročilé žáky. Tento prvek je nepovinný. Proměnná a3 označuje typ předmětu. Jde o předměty povinné, nepovinné a volitelné. Při tvorbě rozvrhu se s těmito možnostmi nepočítá, ale jde o jedno z možných vylepšení. Proměnná a4 označuje seznam vhodných učeben pro výuku předmětu. Lze zde uvést jednu učebnu, ale také seznam všech vhodných učeben. Tento údaj je povinný.

2.3 Učitelé

Jde o informace, které jsou potřebné k přiřazení aprobovaného učitele k jednotlivým předmětům, ale také se zde mohou zadat informace sloužící k dalšímu vylepšení rozvrhu.

Kód:

```
struct Ucitel{
char *a1; //zkratka
char *a2; //jmeno
char *a3; //prijmeni
char *a4; //titul
char *a5; //pohlavi
char *a6; //rodne prijmeni
char *a7; //datum narozeni
char *a8; //místo narozeni
char *a9; //rodne cislo
char *a10; //cislo OP
char *a11; //adresa trvaleho
char *a12; //adresa prechodneho
char *a13; //telefon
char *a14; //email
char *a15; //statni prislusnost
char *a16; //pojistovna
char *a17; //partner
char *a18; //deti
char *a19; //dosazene vzdelani
char *a20; //skoly
char *a21; //praxe
char *a22; //rok nastupu
char *a23; //aprobace
char *a24; //kurzy
Ucitel *next;
};
```

Proměnná a1 označuje zkratku učitele, pod kterou se bude vyučující zobrazovat při tvorbě rozvrhů. Tento údaj je povinný. Další proměnné a2-a22 a a24 jsou nepovinné a jsou zde uvedeny pouze kvůli možným rozšířením programu. Proměnná a23 označuje aprobaci učitele a jako je například MA, IF. Tento údaj je povinný.

2.4 Žáci

Tato databáze není potřeba pro tvorbu samotného rozvrhu, ale může sloužit pro zařazování žáků do jednotlivých tříd a rozdělování ročníků podle různých kritérií jako je například pohlaví.

Kód:

```
struct Zak{
char *a1; //rok nastupu
char *a2; //aktualni trida
char *a3; //jmeno
char *a4; //prijmeni
char *a5; //rodne_cislo
char *a6; //pohlavi
char *a7; //adresa_trvaleho
char *a8; //adresa_prechodneho
char *a9; //pojistovna
char *a10; //datum_narozeni
char *a11; //misto_narozeni
char *a12; //zakony_zastupce
char *a13; //poznamka
char *a14; //OT_jmeno
char *a15; //OT_prijmeni
char *a16; //OT_rodne_cislo
char *a17; //OT_adresa_trvaleho
char *a18; //OT_adresa_prechodneho
char *a19; //OT_telefon
char *a20; //MA_jmeno
char *a21; //MA_prijmeni
char *a22; //MA_rodne_cislo
char *a23; //MA_adresa_trvaleho
char *a24; //MA_adresa_prechodneho
char *a25; //MA_telefon
Zak *next;
```

```
};
```

Žádná ze zadávaných proměnných není při tvorbě rozvrhu použita, ale tyto informace jsou pro vedení školy a třídní učitele důležité v případě potřeby kontaktovat rodiče, nebo například při výpisu vysvědčení, který by se mohl v tomto programu vytvořit jako vylepšení.

2.5 Třídy

Slouží k zadávání jednotlivých tříd a podrobných informací o nich.

Kód:

```
struct Trida{  
char *a1; //zkratka  
char *a2; //nazev tridy  
char *a3; //rok vzniku  
char *a4; //rocnik  
char *a5; //pocet zaku  
char *a6; //kmenova ucebna  
char *a7; //specializace tridy  
char *a8; //tridni ucitel  
char *a9; //zastupce tridniho ucitele  
char *a10; //vypis zaku  
Trida *next;  
};
```

Proměnná a1 je zkratka slouží k jednoznačnému označení třídy. Údaj je povinný a nutný pro tvorbu rozvrhu. Proměnná a4 je ročník třídy ve kterém se nachází. To je potřebné k použití správné dotace hodin, kdy je pro každý ročník uveden přesný počet jednotlivých předmětů.

Ostatní proměnné jsou nepovinné a nejsou tedy při tvorbě rozvrhu nutné. Slouží pro další možná rozšíření programu.

2.6 Učební plán

Slouží k zadávání učebního plánu pro každý ročník zvlášť. Určuje tedy kolik a jakých předmětů se učí v 1. až 9. ročníku.

Kód:

```
struct Plan{  
char *a1; //rocnik  
char *a2; //specializace  
char *a3; //dotace  
Plan *next;  
};
```

Proměnná a1 určuje, o jaký ročník se jedná. Údaj je povinný. Proměnná a2 určuje specializaci ročníku. V jednom ročníku tedy můžou být třídy specializované na matematiku, informatiku, kreslení, atd. a každá z těchto tříd bude mít jinou dotaci hodin. Při generování rozvrhu tento údaj nepoužívám, proto jde o nepovinný údaj. Proměnná a3 určuje, které předměty se v konkrétním ročníku vyučují. Zapisují se takto: 5xMA, 4xCJ, 7xAJ, 2xBI.

2.7 Skupiny

Slouží k jinému rozdělení žáků, nežli podle jednotlivých tříd. Skupiny se používají při tvorbě půlených hodin.

Kód:

```
struct Skupina{  
char *a1; //zkratka  
char *a2; //nazev tridy  
char *a3; //rok vzniku  
char *a4; //pocatecni rocnik  
char *a5; //pocet zaku  
char *a6; //kmenova ucebna  
char *a7; //specializace tridy  
Skupina *next;
```

};

Žádná ze zde použitých proměnných není při tvorbě rozvrhu použita a jsou nepovinné. Slouží zatím pouze pro další rozšíření programu.

2.8 Nutná data pro algoritmus na tvorbu rozvrhu

Pro samotný algoritmus generování rozvrhu jsou nutná především tato data:

Třídy

- označení třídy (1.A)
- ročník, ve kterém se třída nachází (1.)

Učitelé

- zkratka učitele, pod kterou se bude zobrazovat v rozvrhu (CEH)
- předměty, pro které má vyučující aprobaci a může je tedy vyučovat (MA, IF)

Učebny

- zkratka učebny (A01)
- předměty, které je možné v učebně vyučovat (TV)

Plán

- ročník (1.)
- seznam předmětů, které se v daném ročníku vyučují (5xMA, 6xČJ, 3XAJ, 2xTV, atd.)

2.9 Nepovinná data, nutná pro jednoduché rozšíření programu

Ostatní data, která nejsou v mém programu (mimo těch vypsanych v odstavci výše) jsou pro algoritmus nepovinná. Pokud by bylo třeba mohou jednoduše sloužit k zobrazení podrobností u jednotlivých učitelů, žáků, učeben, atd., nebo mohou sloužit jako vstupní data pro další funkce, které by program měl obsahovat.

3. Testovací data

Jako testovací data jsem použil nejprve několik málo smyšlených dat, která sloužila spíše pro kontrolu, zda program opravdu dělá to co má. Poté jsem ale vyhledal na Internetu data pro testování ze ZŠ Na Líše 936/16, Praha 4 – Michle, 141 00. Šlo především o učební plán jednotlivých ročníků a seznam předmětů, které se na škole vyučují. Z prvních smyšlených dat, kdy měl algoritmus vytvořit rozvrh pro zhruba 80 předmětů, jsem tedy přešel na vytváření rozvrhu, kdy se musí umístit zhruba 950 předmětů. Časová náročnost pochopitelně vzrostla ze zhruba 10 vteřin na 3 minuty, ale úspěšnost zadávání se nijak výrazně nezměnila. Jediné, co bylo potřeba při takovémto počtu hodin změnit byl počet učeben a učitelů.

4. Optimalizace

Program jsem se snažil optimalizovat, ale nevyhnul jsem se několikerému projíždění celého seznamu zadávaných hodin a porovnávání s jiným, či se stejným seznamem. To by při delším zkoumání šlo optimalizovat na méně průchodů a více operací během jednoho procházení. Kód programu by se tím stal méně přehledným, ale došlo by k úspoře času jak při generování rozvrhu, tak při výpisu a manuálnímu zadávání rozvrhu. Po mé optimalizaci jsem zmenšil dobu probíhání algoritmu z prvotních 3 minut na zhruba 40 sekund. Šlo především o vynechání některých zbytečných cyklů a nahrazení je cyklem s méně průchody. Dále pak seskupení několika podmínek do jedné a v poslední řadě o kompletní předělání algoritmu, který rozvrh generuje.

5. Generování rozvrhu

Jde o hlavní algoritmus celého programu. Celý rozvrh je zde znázorněn seznamem struktur, kde každá ze struktur má šest položek: den, hodina, předmět, třída, učitel, učebna.

Kód:

```
struct zaznam {  
    char *den; //1-5  
    char *hodina; //1-10  
    char *trida; //1.A  
    char *predmet; //MA  
    char *ucitel; //CEH  
    char *ucebna; //A01  
    zaznam *next;  
}
```

Dokážeme-li pro každý předmět, který se na škole vyučuje určit těchto 6 proměnných, jsou všechny předměty umístěny a přechází se na úpravy, aby například 1. třída neměla žádné odpolední předměty a zbytečné mezery ve výuce.

5.1 Základní úkoly algoritmu

Základními úkoly algoritmu jsou:

- Žádná třída nesmí mít v rozvrhu dva předměty ve stejnou dobu
- Žádný učitel nesmí vyučovat ve dvou třídách zároveň
- V žádné učebně se nesmí vyučovat dva předměty ve stejnou dobu

Další podmínky jsou již nadstandardní a jde u nich spíše o to, aby se neučilo 5 stejných předmětů za sebou ve stejný den, nebo aby jeden vyučující neměl 5 hodin týdně a jiný 50 a aby se nevyučovalo 10 hodin v kuse. Rozvrh, který se generuje není možno okamžitě nasadit do praxe. Vždy je potřeba ho ještě manuálně upravit a dosadit předměty, které algoritmus neumístil do rozvrhu.

5.2 Postup generování rozvrhu

Postup generování rozvrhu má několik fází.

5.2.1 První fáze

V první fázi se ze souborů *Ucitele.dat*, *Ucebny.dat*, *Plan.dat* a *Tridy.dat* načtou potřebné informace do čtyř seznamů. Výsledné seznamy pak vypadají jako v tabulce 1.

Tab. 1: První fáze generování rozvrhu

<i>Ucitele.dat</i>	<i>Ucebny.dat</i>	<i>Plan.dat</i>	<i>Tridy.dat</i>
<i>CEH</i>	<i>A01</i>	<i>1.</i>	<i>1.A</i>
<i>MA</i>	<i>MA</i>	<i>2</i>	<i>1.</i>
<i>CEH</i>	<i>A02</i>	<i>CAS</i>	<i>1.B</i>
<i>ZE</i>	<i>CJ</i>	<i>1.</i>	<i>1.</i>
<i>NOV</i>	<i>A03</i>	<i>9</i>	<i>1.C</i>
<i>MA</i>	<i>ZE</i>	<i>CJ</i>	<i>1.</i>
<i>NOV</i>	<i>A04</i>	<i>1.</i>	<i>2.A</i>
<i>AJ</i>	<i>BI</i>	<i>1</i>	<i>2.</i>
<i>MTY</i>	<i>A05</i>	<i>AJ</i>	<i>2.B</i>
<i>ČJ</i>	<i>TV</i>	<i>Atd.</i>	<i>2.</i>
<i>Atd.</i>	<i>Atd.</i>		<i>Atd.</i>

5.2.2 Druhá fáze

Ve druhé fázi se vytvoří seznam všech předmětů pro všechny třídy na škole a jejich počet. Výsledek po tomto kroku vypadá jako v tabulce 2.

Tab. 2: Druhá fáze generování rozvrhu

186
CJ
125
MA
56
TV
43
VV
41
CAS
28
atd.

5.2.3 Třetí fáze

Ve třetí fázi se do celkového seznamu zadávají časové parametry. Jde tedy o určení dne a hodiny tak, aby se jednotlivé hodiny u jedné třídy nepřekrývaly. Zde je použito podmínek, aby při potřebě zadání více jak jedné hodiny nebyly zadány 2 stejné předměty za sebou.

5.2.4 Čtvrtá fáze

Ve čtvrté fázi se do celkového seznamu přidávají vhodní učitelé, kteří mají potřebnou aprobaci.

5.2.5 Pátá fáze

V páté fázi se do celkového seznamu přidává vhodná učebna, která vyhovuje aprobaci vyučovaného předmětu. V případě, že se nevyhledá vhodná kombinace, procházejí se další varianty hodiny a dne a pokouší se vyhledat vhodnou kombinaci.

5.2.6 Šestá fáze

V šesté fázi probíhá zápis předmětů. Vyhledá-li se vše potřebné, předmět se zapíše do rozvrhu. Tyto poslední fáze se všechny odehrávají v jednom cyklu, takže průběžné výsledky nejsou, ale výsledný seznam vypadá jako v tabulce 3.

Tab. 3: Šestá fáze generování rozvrhu

Den	Hodina	Třída	Předmět	Učitel	Učebna
1	1	1.A	CAS	MAT	B02
2	1	1.A	CAS	MAT	B02
1	2	1.A	CJ	MAT	A02
2	2	1.A	CJ	MAT	A02
3	1	1.A	CJ	MAT	A02
4	1	1.A	CJ	MAT	A02
5	1	1.A	CJ	MAT	A02
0	0	1.A	CJ	0	0
0	0	1.A	CJ	0	0
0	0	1.A	CJ	0	0
0	0	1.A	CJ	0	0
1	3	1.A	AJ	NOV	A06

Nevyhledá-li se vhodná kombinace, parametry předmětu se přepíše na neutrální znak „0“, který používám pro nezařazené předměty. Na příkladu výše je vidět hned několik řádek, kdy program nevyhledal vhodné řešení pro umístění předmětu. Tyto nezařazené předměty se budou muset do rozvrhu zadat manuálně. V šesté poslední fázi se seznam přepisuje do souboru a ukládá se tím na další použití.

5.3 Další možné podmínky při generování rozvrhu

- 1) Jedna základní podmínka je dbát na minimální prostory mezi jednotlivými hodinami u rozvrhu tříd. Je to nejspíše nejobtížnější možná podmínka, kdy je nutno kontrolovat jednotlivé mezery mezi vyučovacími hodinami a v případě příliš mnoha mezer jednotlivé předměty přesouvat.

Příklad špatného rozvrhu:

1.A	1	2	3	4	5	6	7	8
Pondělí	CJ_MAT_A02			CAS_MAT_B02				MA_CEH_A01
Úterý			TV_SPI_A05	CJ_MAT_A02		AJ_NOV_A06		
Středa	CJ_MAT_A02	MA_CEH_A01	VV_POL_B01	CJ_MAT_A02				MA_CEH_A01
Čtvrtek	CAS_HOR_B02	CJ_MAT_A02		HV_SPI_A08			CJ_MAT_A02	
Pátek	CJ_MAT_A02	TV_SPI_A05	CJ_MAT_A02	SP_VOJ_B02		MA_CEH_A01		CJ_MAT_A02

Obr. 1: Příklad špatného rozvrhu

Příklad dobrého rozvrhu:

1.A	1	2	3	4	5	6	7	8
Pondělí	CJ_MAT_A02	MA_CEH_A01	CJ_MAT_A02	CAS_MAT_B02				
Úterý	MA_CEH_A01	CJ_MAT_A02	TV_SPI_A05	CJ_MAT_A02		AJ_NOV_A06		
Středa	CJ_MAT_A02	MA_CEH_A01	VV_POL_B01	CJ_MAT_A02				
Čtvrtek	CAS_HOR_B02	CJ_MAT_A02	MA_CEH_A01	HV_SPI_A08				
Pátek	CJ_MAT_A02	TV_SPI_A05	CJ_MAT_A02	SP_VOJ_B02				

Obr. 2: Příklad dobrého rozvrhu

- 2) Je třeba brát v potaz individuální názor jednotlivých učitelů, kteří například nemohou vyučovat v některý den v týdnu.
- 3) Jednotlivé předměty jsou z hlediska náročnosti různé. Například matematika by neměla být do rozvrhu zadávána na 1. vyučovací hodinu, protože žáci ještě nejsou soustředěni. Zároveň nelze takto těžký předmět do rozvrhu zadat na 7. vyučovací hodinu, kdy už jsou žáci unavení. Dalším speciálním případem je výuka tělesné výchovy, kdy po ní jsou žáci také ovlivněni, takže by měla následovat výuka něčeho méně náročného jako je třeba výtvarná výchova.
- 4) Dále je nutno brát při generování v potaz, že každá třída by měla mít na jeden předmět pouze jednoho vyučujícího.

6. Manuální úpravy rozvrhu

Po vygenerování rozvrhu, kdy algoritmus vytváří jednotlivé prvky rozvrhu, je potřeba manuální úprava, a to kvůli nedostatkům a podmínkám, které algoritmus neobsahuje, nebo by se jimi algoritmus stal příliš složitým a generování by poté nebylo provedeno v reálném čase. Nejčastějším problémem jsou velké mezery mezi jednotlivými hodinami. Tento problém se dá jednoduše korigovat po vygenerování rozvrhu. Toto je poslední část tohoto programu, kdy do téměř vytvořeného rozvrhu se mohou dělat změny a vložit předměty, které algoritmus nedokázal zadat. Jde tu zejména o mazání jednotlivých prvků z rozvrhu a přidávání nových prvků s tím, že je přehledně vidět jak rozvrh učebny, tak i rozvrh učitele a to ještě před tím, než je předmět do rozvrhu zadán. Jde tu o co možná největší názornost, aby se uživatel dokázal v rozvrhu zorientovat.

Úprava rozvrhů je nejdůležitější z hlediska jednotlivých tříd, proto úpravy umožňují pouze v rozvrhu tříd a neumožňují je v rozvrzích učeben ani učitelů. Tam by se při potřebě mohly přidělat, ale program by se musel nechat testovat při běžném provozu.

6.1 Odebírání předmětů z rozvrhu

Odebrání prvku v rozvrhu je nutnou možností v úpravách, která je potřeba například v důsledku nevhodného zadání předmětů hned po sobě, nebo například když není možné umístit do rozvrhu jediný předmět do jediného možného volného místa v rozvrhu. Takováto kolizní situace se nejčastěji řeší vymazáním jiného předmětu, nahrazením tohoto místa předmětem, který jsme chtěli na počátku umístit a posléze nalezením vhodného místa pro předmět, který jsme z rozvrhu vymazali.

6.2 Přidávání předmětů do rozvrhu

Při procházení rozvrhu tříd po projití algoritmem jsou identifikovány předměty, které se nepodařilo do rozvrhu zařadit. Pro tyto předměty je potřeba nalézt vhodné umístění tak, že si zjistíme, který vyučující má kdy volno a která volná a přitom vhodná učebna je ve stejnou dobu dostupná k obsazení. Zde je nutné, aby byly ve stejnou chvíli přehledně zobrazeny tři typy rozvrhů. Zprvė rozvrh třídy, kam chceme hodinu umístit. Dále rozvrh učitele, který má na daný předmět aprobaci či již ve třídě předmět vyučuje. V neposlední řadě rozvrh učebny, která je na výuku vhodná.

7. Možná vylepšení programu:

7.1 Zadávání informací

V programu používám svůj vlastní databázový systém a ukládám data do sedmi různých datových souborů. Prvním a nejspíše i nejvýznamnějším vylepšením by byl přechod na databázi SQL, se kterou by se poté dále pracovalo. Informace by měli být zobrazovány seřazené podle volitelného klíče a díky již vytvořeným funkcím databáze SQL by se dalo s databází lépe pracovat.

7.2 Zadávání informací o učebnách

Je třeba zamezit zadávání stejných zkratk učeben, které slouží jako primární klíč.

Při generování rozvrhu by bylo vhodné zohlednit kapacitu učebny, aby se nestalo, že třída o 30 žácích bude mít přiděleno učebnu pro maximální počet 10 žáků.

Položka specializace by měla být propojena s tabulkou předmětů, které se na škole učí, aby nedocházelo k tomu, že díky překlepu by nešlo zařadit předmět do této třídy. Poté, pomocí klikání tlačítka myši, pouze vybrat ty vhodné předměty, které v dané učebně lze vyučovat.

7.3 Zadávání informací o předmětech

Je potřeba zamezit zadávání stejných zkratk předmětů, které slouží jako primární klíč.

Při tvorbě rozvrhu používat u předmětů typ – povinný, nepovinný, volitelný. Nejdříve by měly být rozvrhovány povinné předměty, poté nepovinné a následně volitelné.

Při zadávání informací o vhodných učebnách by měla existovat možnost vybrat z již zadané tabulky učeben.

7.4 Zadávání informací o učitelích

Je vhodné zamezit zadávání stejných zkratk učitelů, které slouží jako primární klíč.

7.5 Zadávání informací o třídách

Je vhodné zamezit zadávání stejných zkratk tříd, které slouží mimo jiné taky jako primární klíč.

7.6 Konečné návrhy na vylepšení

Vytvořit automatický postup, který umožní správně zadávat vstupní data, aby byla správně provázána a uživatel neopomenul zadat některá důležitá data. Toto by mělo probíhat v novém okně a formou vyplnění dotazníku, který by byl co možná nejvíce „klikací“ a přechod na další či předchozí okno by měl být umožněn pomocí tlačítek další a předchozí. Po několika krocích by uživatel dospěl k hotovému rozvrhu hodin.

V programu není zahrnut výstup na tiskárnu. Ten by měl být možný pro všechny tři typy rozvrhů. Dalším možným výstupem by mohlo být zasílání rozvrhů na e-mailové adresy, které mají jednotliví učitelé.

Celá aplikace by se mohla převést do webového rozhraní a přístup k ní by byl chráněn heslem. Pracovat by na něm poté mohlo i díky využití SQL databáze více uživatelů najednou a například zadávání informací o studentech, což je úkol vhodný pro třídní učitele, by mohli dělat z jakéhokoli počítače s přístupem na Internet.

Aplikace by se dala rozšířit o možnost vkládat půlené hodiny, kdy se třída rozdělí na několik skupiny (obvykle dvě) a každá z těchto skupin by měla jiný rozvrh hodin.

Rozšířit podmínky, které se před generováním rozvrhu mohou zadávat. Zde by se například mohla zadat podmínka jedné volné hodiny při výuce více jak pěti hodin v jednom dni.

Před samotným generováním rozvrhu, které začíná celkovým vymazáním rozvrhu a generováním do prázdného seznamu by mohlo být uživateli umožněno zadat předměty, se kterými by poté algoritmus nehýbal a nechal je zamknuté na svých místech.

Výpis jednotlivých políček v rozvrhu by mohl být barevně odlišen podle klíče, který si zvolí uživatel. Poté by bylo šlo například barevně odlišit jednotlivé předměty, nebo učebny, atd. Barvu jednotlivých prvků by si mohl určit sám uživatel.

Aplikace nyní nepočítá se zadáváním lichého a sudého týdne. Proto by bylo vhodné v další verzi umožnit zadávat souběžně dva rozvrhy najednou, které by se lišili pouze v některých předmětech.

8. Uživatelská příručka

8.1 Instalace programu

Program Deroz – rozvrh hodin je adresář, který obsahuje následující soubory:

- Plan.dat
- Predmety.dat
- Deroz – rozvrh hodin.exe
- Rozvrh2.dat
- Skupiny.dat
- Tridy.dat
- Ucebny.dat
- Ucitele.dat
- Zaci.dat

Adresář můžete umístit kamkoli chcete, ale musí nutně obsahovat všech 8 souborů s příponou dat a jeden soubor s příponou exe. Program spustíte kliknutím na soubor Deroz – rozvrh hodin.exe. Program není nutné instalovat, stačí ho přehrát a je možné ho spustit.

8.2 Úvodní stránka



Obr. 3: Hlavní menu aplikace

Při spuštění programu se vám otevře nové okno s titulkem „Deroz – Rozvrh hodin“. Pod titulkem máte menu, ve kterém se skrývají všechny dostupné funkce programu.

Souhrn jednotlivých názvů v menu:

- Soubor – obecné úkony s datovým souborem obsahující rozvrh hodin a s celou aplikací.
- Data – Zadávání vstupních dat do programu.
- Akce – Generování rozvrhu
- Rozvrh – Zobrazení jednotlivých sestavených rozvrhů
- Zobrazit – slouží k zobrazování logů z programu.

Jednotlivým částem hlavního menu se budeme věnovat v další části uživatelské příručky.

8.3 Soubor



Obr. 4: Hlavní menu aplikace, záložka Soubor

Tato záložka má pod sebou dvě možnosti:

- Nový rozvrh – tato možnost vymaže současný rozpracovaný rozvrh. Slouží především k tomu, abyste mohli manuálně vytvořit rozvrh nový, nebo nový rozvrh vygenerovat.
- Konec – slouží k ukončení aplikace. Každý krok zadávání či odebírání předmětů z rozvrhu je přitom ihned ukládán, takže při opětovném spuštění programu budete moci navázat na předchozí práci. Upozorňujeme, že aplikace „Deroz – Rozvrh hodin“ slouží k vytváření jediného rozvrhu a chcete-li současně vytvářet několik různých variant rozvrhu, nebo několik rozvrhů pro různé školy, je potřeba aplikaci překopírovat do jiné složky, kde budete pracovat na jiné úloze. Doporučení: je dobré si názvy složek pojmenovat přehledně, aby při vytváření nedošlo k záměně a ztrátám dat.

8.4 Zadávání dat

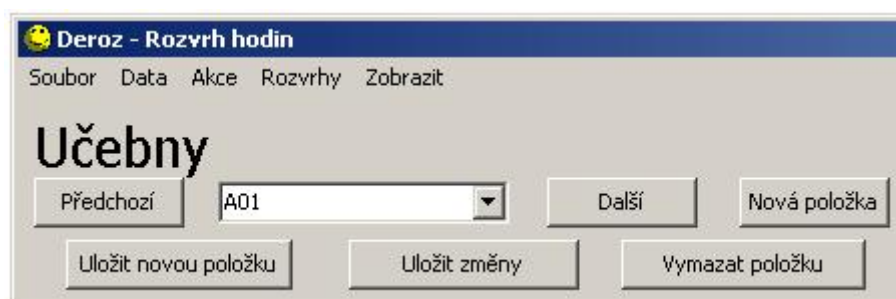
Zadáváním dat se zabývá celá jedna záložka v menu s názvem „Data“.



Obr. 5: Hlavní menu aplikace, záložka Data

Před vytvářením samotného rozvrhu hodin pro jakoukoli školu je potřeba znát některé parametry, které jsou poté při tvorbě rozvrhu nezbytné. Více o jednotlivých typech zadávaných dat v dalších kapitolách.

8.4.1 Tlačítka a funkce při zadávání dat



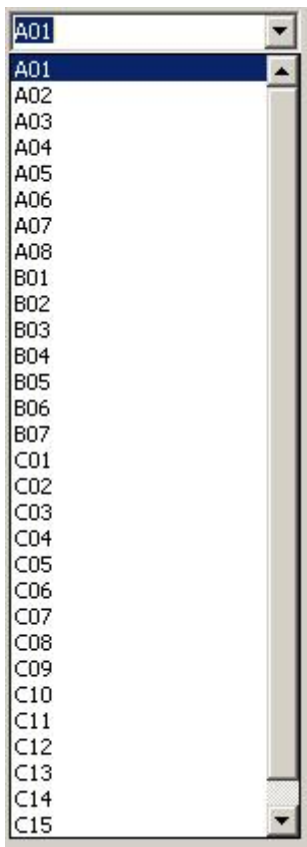
Obr. 6: Popis funkčních tlačítek pro zadávání dat do rozvrhu

Při zadávání všech dat pro generování či manuální zadávání rozvrhu se setkáte se stejnými tlačítky, pomocí kterých můžete provádět operace se zadanými daty. Zde je výpis všech funkčních tlačítek a jejich podrobný popis.

- **Předchozí** – toto tlačítko slouží k zobrazení předchozích dat. Data jsou řazena abecedně. Potřebujete-li se dostat na položku v seznamu, která je abecedně zařazena dříve, použijete toto tlačítko. *Příklad:* Jsme v záložce Učebny a máme zobrazeny informace o učebně A02. Chceme-li zobrazit informace o učebně A01, která je

abecedně před nyní zobrazovanou učebnou A02, klikneme na tlačítko „Předchozí“ a aktuální výpis se změní na parametry učebny A01.

- **ComboBox** (na obrázku výše má v sobě zobrazeno A01) – toto slouží k přehledu všech zadaných prvků v seznamu (viz obrázek níže) a k rychlému přesunu na položku v seznamu, která přímo abecedně nesousedí s aktuálně zobrazenou položkou. To provedeme kliknutím na šipku ukazující směrem dolů, která je na pravé straně tohoto objektu. Po kliknutí se nám zobrazí seznam všech již zadaných položek. V našem případě jde o seznam učeben. Má-li tento seznam příliš mnoho položek, nezobrazí se ihned všechny, ale pro zobrazení položek, které se, jako na obrázku níže, nacházejí vespod seznamu, je nutné použít ScrolBar, což je posuvná lišta na pravém okraji. Posunutím si můžeme nalézt každou ze zadaných položek.



Obr. 7: ScrolBar pro přechod na jinou položku v seznamu

- **Další** – toto tlačítko slouží k zobrazení následujících dat. Data jsou řazena abecedně. Potřebujete-li se dostat na položku v seznamu, která je abecedně zařazena za aktuálním umístěním, použijete toto tlačítko. *Příklad:* Jsme v záložce Učebny a máme zobrazeny informace o učebně A02. Chceme-li zobrazit informace o učebně A03,

kteřá je abecedně za nyní zobrazovanou učebnu A02, klikneme na tlačítko „Další“ a aktuální výpis se změní na parametry učebny A02.

- **Nová položka** – toto tlačítko slouží pro vymazání všech polí pro vkládání dat a přípravu pro zadávání dat u nové položky. Chcete-li vytvořit nový prvek v seznamu, můžete pomocí tohoto tlačítka vymazat všechny políčka a připravit tak formulář na vkládání vašich nových dat.
- **Uložit novou položku** – toto tlačítko slouží k uložení zadaných dat do seznamu. Vyplníme-li tedy potřebná políčka ve formuláři, je potřeba tyto data ještě uložit a tím je zapsat do seznamu. Neuložíte-li novou položku, o zapsaná data při další akci přijdete. Toto tlačítko lze také použít pro vkládání podobných prvků do seznamu jako ty, které již v seznamu jsou. *Příklad:* budeme chtít zadat do seznamu novou učebnu F01, která je téměř ve všech parametrech stejná jako učebna A01. Zobrazíme si tedy učebnu A01, změníme data, která jsou odlišná od učebny A01, například název a zkratka učebny a poté pomocí tlačítka „Uložit novou položku“ vložíme novou položku do našeho seznamu učeben.
- **Uložit změny** – toto tlačítko slouží k úpravám jednotlivých položek v seznamu. Dojde-li tedy k nějaké změně v některých částech položky v seznamu, můžeme je upravit a pomocí tohoto tlačítka změnit. *Příklad:* Pro učebnu A01 máme zadanou specializaci na matematiku, ale jelikož dojde k přestavbě učebny na učebnu výtvarné výuky, potřebujeme tuto změnu promítnout i do programu, který bude rozvrh generovat. Proto si najedeme pod „Menu >> Data >> Učebny“, zde si pomocí ComboBoxu, nebo tlačítek „předchozí“ a „další“ najedeme na učebnu A01, přepíšeme v řádku Specializace hodnotu „MA“ na „VV“ a kliknutím na tlačítko „Uložit změny“ se nám tato položka seznamu změní. Změnu již nelze vzít zpět.
- **Vymazat položku** – toto tlačítko slouží pro smazání jednotlivých položek v seznamu. Při používání tohoto tlačítka buďte opatrní, protože změny již nelze vrátit zpátky. *Příklad:* Jedna z učeben B05, které dříve sloužili pro výuku, se změnila na kabinet. Potřebujeme tedy tuto učebnu odstranit ze seznamu, aby se s ní při tvorbě rozvrhu nepočítalo. Proto si najedeme pod „Menu >> Data >> Učebny“, zde si pomocí ComboBoxu, nebo tlačítek „předchozí“ a „další“ najedeme na učebnu B05 a zmáčknutím tlačítka „Vymazat položku“ ji odstraníme ze seznamu.

8.4.2 Učebny

The screenshot shows a software window titled "Deroz - Rozvrh hodin". The menu bar includes "Soubor", "Data", "Akce", "Rozvrhy", and "Zobrazit". The main title is "Učebny". Below the title are several buttons: "Předchozí", a dropdown menu showing "A01", "Další", "Nová položka", "Uložit novou položku", "Uložit změny", and "Vymazat položku". The form contains four labeled input fields: "Zkratka(*)" with value "A01", "Název učebny" with value "Matematika", "Kapacita(*)" with value "30", and "Specializace(*)" with value "MA". A note at the bottom right states "* - povinné údaje pro tvorbu rozvrhu".

Obr. 8: Zadávání dat do databáze učeben

Pro generování či manuální zadávání hodin do rozvrhu je potřeba mít alespoň jednu učebnu, ve které se může(mohou) předmět(y) vyučovat. K zadávání učeben slouží právě tato záložka, do které se dostanete přes „Menu >> Data >> Učebny“.

Povinné údaje, které musíte vyplnit, jsou zde označeny *.

Popis jednotlivých údajů:

- Zkratka – označení učebny, které se bude používat při výpisu rozvrhu. Doporučují se 3 znaková označení.
- Název učebny – zde můžete vypsát podrobnosti (co se v dané učebně vyučuje, nebo libovolný jiný popis, který vám pomůže v orientaci).
- Kapacita – zde se zadává maximální počet žáků, které může učebna pojmout při výuce. *Příklad: Malé učebny jsou pro 10 osob, zatímco tělocvična bude mít kapacitu nastavenou na 30 a více osob.*
- Specializace – označuje jaký předmět je pro výuku v konkrétní učebně možno učit.

Je-li učebna vhodná pro výuku více předmětů, vypisují se předměty oddělené čárkou a mezerou. Příklad: Učebna A01 je vybavena pro výuku matematiky, takže specializace bude MA. Učebna A02 je univerzální učebnou, která je vhodná pro výuku více předmětů a proto zde bude vypsáno: CAS, CJ, AJ, MA, VV, HV, SP, IF, BI, FY, ZE, DE, VO, VZ, CH

8.4.3 Předměty

Deroz - Rozvrh hodin
Soubor Data Akce Rozvrhy Zobrazit

Předměty

Předchozí MA Další Nová položka

Uložit novou položku Uložit změny Vymazat položku

Zkratka(*)	MA
Název předmětu	Matematika
Typ	Povinný
Vhodné místnosti(*)	MA

* - povinné údaje pro tvorbu rozvrhu


Obr. 9: Zadávání dat do databáze předměty

Zde musíte zadat jednotlivé předměty, které se na škole vyučují.

Popis jednotlivých údajů:

- Zkratka – je označení vyučovaného předmětu. Doporučujeme zadávat 2-3 znaková označení pro každý předmět. Příklad: MA pro Matematika, CJ pro český jazyk, atd.
- Název předmětu – zde je možné předmět popsat a napsat si k němu například nějakou poznámku. To se může hodit například při zadávání předmětů jako třeba AJ1 – Anglický jazyk začátečníci, AJ2 – Anglický jazyk pokročilí, atd.
- Typ – označení, je-li předmět povinný, volitelný či nepovinný
- Vhodná místnost – zde je seznam specializací místností, které jsou pro výuku vhodné. Může zde být jedna specializace, nebo více oddělených čárkou a mezerou. Například: MA, CJ, BI, ZE.

8.4.4 Učitelé

 **Deroz - Rozvrh hodin**

Soubor Data Akce Rozvrhy Zobrazit

Učitelé

Předchozí

CEH

Další

Nová položka

Uložit novou položku

Uložit změny

Vymazat položku

Zkratka(*)	CEH
Jméno	Jan
Příjmení	Cehák
Titul(y)	
Pohlaví	M
Rodné příjmení	
Datum narození	9.3.1983
Místo narození	Praha
Rodné číslo	8303090444
Číslo OP	933102929
Adresa trvalého bydliště	Dědinská 897, Praha 6, 161 00
Adresa přechodného bydliště	
Telefon	605940011
E-mail	jan.cehak1@seznam.cz
Státní příslušnost	České
Zdravotní pojištění	Všeobecná zdravotní pojišťovna
Jméno partnera	Kateřina Ceháková
Počet dětí	0
Dosažené vzdělání	SŠ
Absolvované školy	Gymnázium Arabská
Praxe před nástupem	
Rok nástupu	2009
Aprobace pro předměty(*)	MA, ZE
Absolvované kurzy	MS Windows

* - povinné údaje pro tvorbu rozvrhu


Obr. 10: Zadávání dat do databáze učitelé

Zde jsou veškeré potřebné informace o jednotlivých učitelích. Některé jsou potřebné pro tvorbu rozvrhu, jiné jsou třeba pro možná rozšíření programu, kterým zde věnuji samostatnou kapitolu.

Popis jednotlivých údajů:

- Zkratka – unikátní označení učitele, pod kterým se bude zobrazovat ve vytvořených rozvrzích.
- Jméno, Příjmení, Titul, Pohlaví, Rodné příjmení, Datum narození, Místo narození, Rodné číslo, Číslo OP, Adresa trvalého bydliště, Adresa přechodného bydliště, Telefon, E-mail, Státní příslušnost, Zdravotní pojištění, Jméno partnera, Počet dětí, Dosažené vzdělání, Absolvované školy, Praxe před nástupem, Rok nástupu, Absolvované kurzy – všechny tyto údaje jsou nepovinné a slouží spíše k dodatečným informacím o učitelích, nežli k tvorbě rozvrhu jako takového.
- Aprobace pro předměty – zde je nutné uvést zkratky předmětů, na které má vyučující aprobaci. Je- předmětů více, oddělují se čárkou a mezerou.

8.4.5 Žáci

 **Deroz - Rozvrh hodin**

Soubor Data Akce Rozvrhy Zobrazit

Žáci

Předchozí

2009

Další

Nová položka

Uložit novou položku

Uložit změny

Vymazat položku

Rok nástupu	2009
Třída (aktuální)(*)	1.A
Jméno	Alena
Příjmení(*)	Nádeníčková
Rodné číslo	010101/0005
Pohlaví	Ž
Adresa trvalého bydliště	Pražská 15, Liberec 15, 999999
Adresa přechodného bydliště	
Pojišťovna	Všeobecná zdravotní pojišťovna
Datum narození	1.1.2001
Místo narození	Liberec
Zákonný zástupce	Pavel Nádeníček
Poznámka	Alergie na křidu
Matka: jméno	Pavel
Matka: příjmení	Nádeníček
Matka: rodné číslo	010101/0005
Matka: Adresa trvalého bydliště	Pražská 15, Liberec 15, 999999
Matka: Adresa přechodného bydliště	
Matka: Telefon	123123123
Otec: jméno	Věra
Otec: příjmení	Nádeníčková
Otec: rodné číslo	010101/0005
Otec: Adresa trvalého bydliště	Pražská 15, Liberec 15, 999999
Otec: Adresa přechodného bydliště	
Otec: Telefon	777123456

* - povinné údaje pro tvorbu rozvrhu

Obr. 11: Zadávaní dat do databáze žáci

V současné verzi programu není žádný z těchto údajů používán, ale je možné je použít v možných vylepšeních programu.

8.4.6 Třídy

Deroz - Rozvrh hodin
Soubor Data Akce Rozvrhy Zobrazit

Třídy

Předchozí 1.A Další Nová položka

Uložit novou položku Uložit změny Vymazat položku

Zkratka(*)	1.A
Název třídy	
Rok vzniku(*)	
Počáteční ročník(*)	1.
Počet žáků	
Kmenová učebna	
Specializace třídy	
Třídní učitel	
Zástupce třídního učitele	
Výpis žáků	

* - povinné údaje pro tvorbu rozvrhu

Obr. 12: Zadávání dat do databáze třídy

Tato databáze slouží k ukládání jednotlivých tříd. Nebude-li zde třída uvedena, nebude program moci pro ni vytvořit rozvrh.

Popis jednotlivých údajů:

- Zkratka – zkratka třídy, pod kterou se bude třída zobrazovat ve vytvořeném rozvrhu
- Název třídy – zde se můžou vložit různé upřesňující údaje o třídě.
- Rok vzniku – zatím není v programu použito, ale tento údaj je nutný k automatickému posunu tříd z jednoho ročníku do vyššího, ke kterému dochází během letních prázdnin. V případě, že tento údaj nebude používán, bude se muset vše dělat manuálně.
- Počáteční ročník – slouží k tomu, když například v 5. ročníku vznikne nová třída, která ještě na škole nebyla. Praktické využití je například při velkém přílivu studentů.
- Počet žáků – tento údaj je důležitý pro možné rozšíření programu a to především ke kontrole kapacity učebny při vytváření rozvrhu.
- Kmenová učebna – slouží ke specifikaci, která učebna třídě patří a o kterou se má třída starat.
- Specializace třídy – zde by mělo být uvedeno, o jakou třídu se jedná. Je-li třída například se zaměřením na sport, matematiku, informatiku, atd.

- Třídní učitel – tento údaj je důležitý pro rychlé vyhledávání v databázi žáků a jejich třídních učitelů.
- Zástupce třídního učitele – zástup při absenci třídního učitele.
- Výpis žáků – zde by měl být seznam všech žáků, které třída má.

8.4.7 Skupiny

Deroz - Rozvrh hodin

Soubor Data Akce Rozvrhy Zobrazit

Skupiny žáků

Předchozí AJ11 Další Nová položka

Uložit novou položku Uložit změny Vymazat položku

Zkratka(*)	AJ11
Název skupiny	Anglický jazyk začátečníci
Maximální počet žáků	15
Aktuální počet žáků	0
Část tříd(y)(*)	1.A
Výčet členů	
Disjunktní skupiny(*)	AJ12

* - povinné údaje pro tvorbu rozvrhu

Obr. 13: Zadávání dat do databáze žáků

Tato databáze je nutná pro rozdělení tříd do jednotlivých skupin, je-li potřebné třídu rozdělit například podle úrovně jazykové znalosti, nebo podle pohlaví na tělesnou výchovu, atd.

Tato databáze je připravena na další rozšíření programu a nyní není žádný z jejích parametrů důležitý pro tvorbu rozvrhu.

8.4.8 Učební plán

The screenshot shows a software window titled "Deroz - Rozvrh hodin". The menu bar includes "Soubor", "Data", "Akce", "Rozvrhy", and "Zobrazit". The main title is "Učební plán". Below the title are several buttons: "Předchozí", a dropdown menu showing "1.", "Další", and "Nová položka". Below these are three more buttons: "Uložit novou položku", "Uložit změny", and "Vymazat položku". The form has three input fields: "Ročník(*)" with "1.", "Specializace(*)" with "žádná", and "Dotace(*)" with "2xCAS, 9xCJ, 1xAJ, 4xMA, 1xVV, 1xHV, 2xTV, 1xSP". A note at the bottom right states: "* - povinné údaje pro tvorbu rozvrhu".

Obr. 14: Zadávání dat do databáze učebního plánu

Toto je jedna z nejdůležitějších databází v programu. Zde se musí každému ročníku určit přesná týdenní dotace hodin, kterou bude ročník mít.

Popis jednotlivých údajů:

- Ročník – Zde se zadává ročník, pro který dotaci hodin zadáváme. Možné hodnoty jsou například: 1., 2., 9.
- Specializace – Tento údaj je zde uveden proto, že ne v každém ročníku jsou stejné třídy se stejnou hodinovou dotací. Příklad: sportovní třída bude mít více hodin tělesné výchovy týdně a naopak třída se zaměřením na matematiku bude mít hodin tělocviku méně.
- Dotace – zde se vypisují jednotlivé předměty a jejich počet za týden. Formát zápisu je: [počet]x[předmět], [...]

8.5 Generování rozvrhu

Jakmile jsou zadána všechna data potřebná pro tvorbu rozvrhu, je možné přejít ke generování rozvrhu. To provedete kliknutím na „Menu >> Akce >> Generovat rozvrh“



Obr. 15: Menu - Akce - Generování rozvrhu

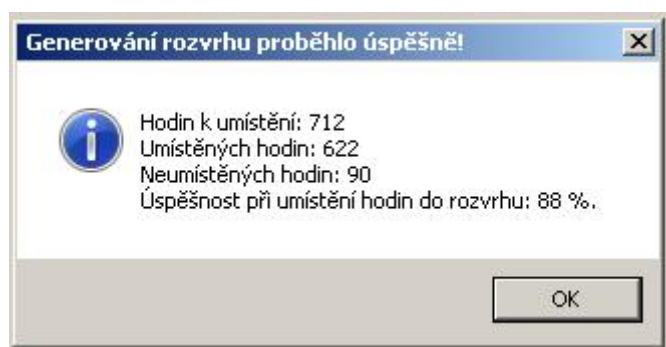
Po kliknutí se vám zobrazí následující stránka:



Obr. 16: Zadávání podmínek pro generování rozvrhu

Generování je automatické a vyžaduje zadání pouze jediné proměnné a to maximálního počtu možných vyučovacích hodin v jednom dni. Jestliže tuto hodnotu ne zadáte, je zde přednastaveno 8 hodin.

Poté již jen stačí kliknout na tlačítko „Odeslat požadavky“ a rozvrh se začne generovat. Doba generování rozvrhu je závislá na počtu předmětů, které má program umístit a na rychlosti počítače, na kterém je program spuštěn. Pro „klasickou“ školu s 9 ročníky a 3 třídy v každém ročníku a maximální hodinovou dotací trvá generování zhruba 40 sekund. Po ukončení generování se vám zobrazí následující informační panel, který vám oznámí, zda bylo generování úspěšné a další podrobnosti o úspěšnosti umísťování předmětů:



Obr. 17: Výsledná tabulka po dokončení generování rozvrhu

Úspěšnost umístění hodin do rozvrhu je závislá především na dvou faktorech, které můžeme ovlivnit.

- Dostatek učitelů, kteří vyučují daný předmět
- Dostatek učeben, které jsou vhodné pro umístění předmětu

Obě tyto podmínky jsme schopni ovlivnit a v další fázi tvorby rozvrhu zjistíme, jestli je jedna z těchto věcí příčinou nedostatečného umístění předmětů či nikoli.

8.6 Konečné úpravy v rozvrhu

Na rozvrh se nyní můžeme podívat a zjistit, jestli nám algoritmus vytvořil rozvrh přijatelný a my v něm budeme muset udělat pouze drobné změny, nebo rozvrh zcela nevhodný a v tom případě v něm budeme muset dělat změny zásadní.

Jak vypadá samotný rozvrh se podíváme do záložky „Menu >> Rozvrhy“. Nabídka zde vypadá následovně:



Obr. 18: Menu - Rozvrhy

Zde nalezneme 3 možnosti, které můžeme zvolit. Každá nám zobrazí výsledek vygenerovaného rozvrhu.

8.6.1 Třidy

Klikneme-li na tuto možnost, zobrazí se nám následující rozvrh tříd.

Výpis rozvrhu tříd									Nezařazené předměty	
Předchozí	1.A	Další							1	CJ
1.A	1	2	3	4	5	6	7	8	2	CJ
Pondělí	CAS_MAT_B02	CJ_MAT_A02	AJ_NOV_A06	MA_CEH_A01	VV_POL_B01	HV_SPI_A08	TV_SPI_A05		3	CJ
Úterý	SP_VOJ_B02								4	CJ
Středa	CAS_MAT_B02	CJ_MAT_A02	MA_CEH_A01	TV_SPI_A05					5	CJ
Čtvrtek									6	CJ
Pátek	CJ_MAT_A02	MA_CEH_A01							7	MA
									8	
									9	
									10	
									11	

Obr. 19: Výpis rozvrhu tříd

Podobně jako při zadávání vstupních dat i zde můžeme pomocí dvou tlačítek „Předchozí“ a „Další“ přeskakovat z aktuálně zobrazeného rozvrhu třídy na třídy předchozí a následující. Mimo toho můžeme také použít **ComboBox**, který je umístěn mezi těmito tlačítky a pomocí něhož můžeme zobrazit rozvrh, který je od aktuálně zobrazeného vzdálen.

Pomocí tlačítka **ComboBox** se také můžeme podívat, kolik máte celkem tříd, u kterých je rozvrh vygenerován.

Vpravo vedle rozvrhu máme seznam nezařazených předmětů, které se ještě musí do rozvrhu zadat, aby byl kompletní.

Jsme-li v této sekci, můžeme, na rozdíl od dvou zbývajících sekcí, provádět manuální úpravy v rozvrhu.

Mazání

Mazání záznamu v rozvrhu provádíme následovně. Klikneme na obdélník, ve kterém se nežádoucí předmět nachází a vyskočí nám následující otázka:



Obr. 20: Potvrzení smazání zadaného předmětu

Klikneme-li na „Ano“, záznam bude smazán. Klikneme-li na „Ne“, žádná změna nenastane.

Přidávání nového záznamu do rozvrhu

Přidávání záznamu se provádí například v případě, že ještě nemáme umístěny všechny hodiny v rozvrhu, jako tomu je u obrázku na předchozí straně.

Nejprve si musíme vybrat, kam chceme umístit předmět a klikneme na daný obdélník. Je-li v obdélníku již umístěn nějaký předmět, budeme dotázáni na jeho vymazání. Nebude-li v obdélníku umístěn žádný předmět, zobrazí se nám vpravo vedle tabulky s nezařazenými předměty tabulka podobná, ze které ale již můžeme vybírat, který předmět chceme na dané místo umístit. Navíc se vedle této tabulky ještě zobrazí další 4 menší tabulky, ve kterých se zatím nic nezobrazuje. Pravá strana našeho hlavního panelu poté bude vypadat následovně:

Nezařazené předměty

			Obsazení učitelé:	Obsazené učebny:
1	CJ			
2	CJ	CJ		
3	CJ	CJ		
4	CJ	CJ		
5	CJ	CJ		
6	CJ	CJ		
7	MA	CJ		
8		MA		
9				
10				
11			Volní učitelé:	Volné učebny:
12				
13				
14				
15				
16				
17				
18				
19				

Obr. 21: Výběr nezařazených předmětů

Vybereme-li jeden z předmětů v druhé tabulce kliknutím, vyplní se nám 4 menší tabulky zkratkami učitelů a místností, které jsou vhodné pro výuku daného předmětu. Všechny šest tabulek pak bude vypadat následovně:

Nezařazené předměty

			Obsazení učitelé:	Obsazené učebny:
1	CJ			
2	CJ	CJ	MAT	A02
3	CJ	CJ	NO3	B02
4	CJ	CJ	NO4	B03
5	CJ	CJ	NO5	B04
6	CJ	CJ	NO6	B05
7	MA	CJ	NO8	B06
8		MA	NO9	B07
9			NO10	C01
10				
11			Volní učitelé:	Volné učebny:
12				C12
13				C13
14				C14
15				C15
16				
17				
18				
19				

Obr. 22: Manuální zadávání předmětu do rozvrhu hodin

Jak je vidět na obrázku, tak pro tento den a hodinu již nemáme učitele, který by mohl předmět CJ vyučovat. Nyní máme na výběr ze dvou možností:

- Změnit den, změnit hodinu, změnit den i hodinu kdy se předmět má umístit.
- Najít si rozvrh jednoho z učitelů a jeden z předmětů, který se kryje s tímto vymazat.

Budeme postupovat podle první možnosti a s tou nám ještě pomůže následující funkce, kterou program umožňuje. Klikneme-li na nějakého ze zobrazených učitelů či učeben, zobrazí se nám jejich rozvrh. Nyní tedy klikneme na učitele MAT a na učebnu C12 a zobrazí se nám tři rozvrhy. Prvním bude náš rozvrh třídy, do kterého se snažíme zadat předmět CJ, druhým bude rozvrh učitele se zkratkou MAT a třetím bude rozvrh učebny se zkratkou C12. Levá část hlavního panelu poté bude vypadat následovně:

1.A	1	2	3	4	5	6	7	8
Pondělí	CAS_MAT_B02	CJ_MAT_A02	AJ_NOV_A06	MA_CEH_A01	VV_POL_B01	HV_SPI_A08	TV_SPI_A05	
Úterý	SP_VOJ_B02							
Středa	CAS_MAT_B02	CJ_MAT_A02	MA_CEH_A01	TV_SPI_A05				
Čtvrtek								
Pátek	CJ_MAT_A02	MA_CEH_A01						
MAT	1	2	3	4	5	6	7	8
Pondělí	CAS,1.A,B02	CJ,1.A,A02	CAS,2.A,B04	CJ,2.A,A02	CJ,2.A,A02	CJ,2.A,A02	CJ,2.A,A02	
Úterý	CJ,2.A,A02	CJ,2.A,A02	CJ,2.A,A02	CJ,3.C,A02	CJ,3.C,A02	CJ,4.C,A02	CJ,4.C,A02	
Středa	CAS,1.A,B02	CJ,1.A,A02	CAS,2.A,B04	CAS,2.C,B03	CAS,3.A,B03	CJ,7.C,A02	CJ,7.C,A02	
Čtvrtek	CAS,3.C,B07	CAS,4.B,B03	CAS,5.A,B03	CAS,5.C,B03				
Pátek	CJ,1.A,A02							
C12	1	2	3	4	5	6	7	8
Pondělí	DE,7.BTOP	VO,7.BTOP	ZE,6.CXAO	HV,8.CNO7	AJ,6.CNO2	DE,7.ATOP	DE,8.ATOP	
Úterý	BI,9.CFIL	AJ,9.ANO2	ZE,8.BXAO			VZ,8.BVAC		
Středa	VZ,8.CVAC	AJ,9.BČEP	DE,9.ATOP					
Čtvrtek	VZ,7.BVAC							
Pátek								

Obr. 23: Rozvrhy třídy, učitele a učebny pro lepší orientaci v manuálním zadávání předmětu do rozvrhu hodin

Nyní tedy můžeme vybrat místo, které je pro umístění vhodné. V tomto případě by se nám mohlo jako nejlepší řešení zdát umístění předmětu v pátek 3. vyučovací hodinu. Pomocí klikání ve všech funkčních polích pochopitelně můžeme nalézt i vhodnější umístění pro předmět. Navíc můžeme i některé předměty vymazat, atd. Od tohoto bodu již vše závisí pouze na uživateli programu, který může mazat, umísťovat a volit z velkého množství různých kombinací.

8.6.2 Učitelé

Dalším rozvrhem, který si můžeme zobrazit, je rozvrh učitelů. Na výpis se dostaneme jednoduše pomocí kliknutí na „Menu >> Rozvrhy >> Učitelé“ Zobrazí se vám poté takovýto rozvrh:

Výpis rozvrhu učitelů								
Předchozí	SPI		Další					
SPI	1	2	3	4	5	6	7	8
Pondělí	HV,8.A,A08	TV,8.A,A05	HV,7.A,A08	HV,7.B,A08	HV,1.C,A08	HV,1.A,A08	TV,1.A,A05	
Úterý	TV,1.B,A05	TV,1.C,A05	TV,4.C,A05	HV,7.C,A08	TV,9.B,A05	HV,2.C,A08	HV,2.A,A08	
Středa	TV,5.B,A05	TV,2.A,A05	TV,2.C,A05	TV,1.A,A05	TV,2.B,A05	TV,2.D,A05	TV,3.A,A05	
Čtvrtek	TV,1.B,A05	TV,1.C,A05	TV,3.B,A05	TV,3.C,A05	TV,4.A,A05	TV,4.B,A05	TV,4.C,A05	
Pátek	TV,5.A,A05	TV,2.A,A05	TV,2.B,A05	TV,2.C,A05	TV,2.D,A05	TV,3.A,A05	TV,5.B,A05	

Obr. 24: Příklad rozvrhu učitele se zkratkou SPI

Jak je zde vidět, tak učitelů tělocviku a hudební výchovy je nejspíše na škole, kterou zde simulujeme, málo. Toto je jedna z věcí, které při pohledu na tyto rozvrhy můžeme zjistit a dále podle toho jednat.

8.6.3 Učebny

Dalším a přitom posledním typem rozvrhu, který zde máme možnost zobrazit je rozvrhem učeben. Každá učebna má svůj vlastní rozvrh a stejně jako u rozvrhu učitelů v něm nemůžeme dělat žádné změny. Jeden z rozvrhů učeben vypadá například takto:

Výpis rozvrhu učeben								
Předchozí	A01		Další					
A01	1	2	3	4	5	6	7	8
Pondělí	MA 5.B CEH	MA 5.C CEH		MA 1.A CEH	MA 7.A CEH		MA 7.B CEH	
Úterý	MA 4.C CEH	MA 6.B CEH	MA 8.C NO1	MA 7.A CEH	MA 2.A CEH	MA 4.A CEH	MA 3.A CEH	
Středa	MA 2.A CEH	MA 7.C CEH	MA 1.A CEH	MA 6.B CEH	MA 5.A CEH	MA 4.C CEH		
Čtvrtek	MA 2.A CEH	MA 3.C CEH	MA 7.C CEH		MA 8.A CEH		MA 9.C CEH	
Pátek	MA 2.A CEH	MA 1.A CEH	MA 6.C CEH	MA 9.A CEH				

Obr. 25: Příklad rozvrhu učebny s označením A01

Závěr

Po celou dobu, co jsem program vytvářel, jsem věděl, že vytvořit algoritmus, který by vytvořil kompletní rozvrh hodin pro danou školu a nemuselo by se do něj poté vůbec zasahovat, zatím neexistuje a v některých případech, kdy je na škole například nedostatek učeben je přímo nemožné ho vytvořit. U každého algoritmu na sestavování rozvrhu hraje klíčovou roli člověk, který na konci provede úpravy, kterými rozvrh předělá do podoby, se kterou se poté celý rok vyučuje. Milým překvapením pro mne tedy bylo, když jsem zjistil, že mi algoritmus „zarozvrhuje“ zhruba 90% všech předmětů. Program by se sice již v této podobě dal použít pro tvorbu rozvrhu, ale před samotným uvedením do praxe by ještě vyžadoval řadu zásahů, které by si vyžádaly spoustu času a testování přímo s lidmi, kteří jsou již v této problematice znalí.

Program svou velikostí a náročností operací, kdy se například musí během jednoho algoritmu pracovat s šesti různými databázemi, předčil moje očekávání a v době vytváření hlavního algoritmu, který sice není dlouhý, co se kódu týče, ale spíše náročný, co se logiky týče, jsem prožil několik bezesných nocí přemýšlením, kde se v tom kódu nachází ona chyba.

Program je nyní ve fázi „release“ a množství funkcí, které jsem do něj chtěl zapracovat, bohužel zůstalo pouze na papíře. V této diplomové práci je můžete nalézt v kapitole „možná vylepšení programu“.

Program nyní splňuje základní kritéria, která jsem od něho očekával, jako například vyřešení základních problémů: v jedné učebně se smí v jeden okamžik vyučovat maximálně jeden předmět, jeden vyučující může učit v jeden okamžik maximálně jeden předmět a jedna třída může mít v jeden okamžik pouze jediný předmět a dále program splňuje několik doplňujících podmínek jako je maximální počet hodin v jednom dni, snaha vyhnout se zařazení dvou stejných hodin ihned za sebe, atd. Práce na tomto projektu pochopitelně není zdaleka u konce a možností, jak projekt rozšířit a vylepšit je až příliš mnoho na to, abych je mohl všechny zpracovat sám a přitom tomu nevěnoval několik měsíců až let.

Literatura

STROUSTRUP Bjarne –*The C++ programming language. Second edition.* Addison-Wesley Publishing Company. 1990. 691 s. ISBN 0-201-51459-1.

Příloha 1: Seznam obrázků

Obr. 1: Příklad špatného rozvrhu	23
Obr. 2: Příklad dobrého rozvrhu	23
Obr. 3: Hlavní menu aplikace	28
Obr. 4: Hlavní menu aplikace, záložka Soubor.....	29
Obr. 5: Hlavní menu aplikace, záložka Data	30
Obr. 6: Popis funkčních tlačítek pro zadávání dat do rozvrhu.....	30
Obr. 7: ScrolBar pro přechod na jinou položku v seznamu.....	31
Obr. 8: Zadávání dat do databáze učeben.....	33
Obr. 9: Zadávání dat do databáze předměty	34
Obr. 10: Zadávání dat do databáze učitelé.....	35
Obr. 11: Zadávání dat do databáze žáci.....	37
Obr. 12: Zadávání dat do databáze třídy.....	38
Obr. 13: Zadávání dat do databáze žáků.....	39
Obr. 14: Zadávání dat do databáze učebního plánu	40
Obr. 15: Menu - Akce - Generování rozvrhu.....	41
Obr. 16: Zadávání podmínek pro generování rozvrhu	41
Obr. 17: Výsledná tabulka po dokončení generování rozvrhu.....	42
Obr. 18: Menu - Rozvrhy.....	43
Obr. 19: Výpis rozvrhu tříd.....	43
Obr. 20: Potvrzení smazání zadaného předmětu.....	44
Obr. 21: Výběr nezařazených předmětů	45
Obr. 22: Manuální zadávání předmětu do rozvrhu hodin.....	46
Obr. 23: Rozvrhy třídy, učitele a učebny pro lepší orientaci v manuálním zadávání předmětu do rozvrhu hodin.....	47
Obr. 24: Příklad rozvrhu učitele se zkratkou SPI.....	48
Obr. 25: Příklad rozvrhu učebny s označením A01	48

Příloha 2: Seznam tabulek

Tab. 1: První fáze generování rozvrhu	20
Tab. 2: Druhá fáze generování rozvrhu	21
Tab. 3: Šestá fáze generování rozvrhu.....	22

Příloha 3: Obsah CD

Přiložené CD má následující strukturu souborů

- Deroz – Rozvrh hodin
 - Deroz – Rozvrh hodin.exe
 - Plan.dat
 - Předmět.dat
 - Rozvrh2.dat
 - Skupiny.dat
 - Tridy.dat
 - Učebny.dat
 - Ucitele.dat

- Zaci.dat
- Diplomová práce
 - Deroz – Rozvrh hodin.pdf
- Zdrojový kód
 - Plan.cpp
 - Plan.dat
 - Plan.h
 - Predmety.cpp
 - Predmety.dat
 - Predmety.h
 - Project1.cbproj
 - Project1.cbproj.local
 - Project1.cpp
 - Project1.res
 - ProjectGroup1.groupproj
 - ProjectGroup1.groupproj.local
 - ProjectGroup1_prjgroup.tvconfig
 - Release.optset
 - Roz1.cpp
 - Roz1.h
 - Rozvrh2.dat
 - Roz_man.cpp
 - Roz_man.h
 - Roz_zobraz.cpp
 - Roz_zobraz.h
 - Roz_zobraz_tridy2.cpp
 - Roz_zobraz_tridy2.h
 - Roz_zobraz_ucitele.cpp
 - Roz_zobraz_ucitele.h
 - Sdiapp.cbproj.local
 - Skupiny.cpp
 - Skupiny.dat
 - Skupiny.h
 - Tridy.cpp
 - Tridy.dat
 - Tridy.h
 - Ucebny.cpp
 - Ucebny.dat
 - Ucebny.h
 - Ucitele.cpp
 - Ucitele.dat
 - Ucitele.h
 - Unit2.cpp
 - Unit2.dfm
 - Unit2.h
 - Viditelnost.cpp
 - Viditelnost.h
 - Zaci.cpp
 - Zaci.dat
 - Zaci.h